Amendments to the Specification

Please replace the specification of record with the substitute specification enclosed herewith.

PREDICTIVE PREDOWNLOAD OF TEMPLATES WITH DELTA ENCODING

RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Patent Application Serial No. 09/734,910, filed 11 December 2000, entitled "Predictive Pre-download Using Normalized Network Object Identifiers."

[0002] This application is related to U. S. Patent Application Serial No. 10/058,232, filed 19 October 2001, entitled "Differential Caching with Many-to-One and One-to-Many Mapping."

BACKGROUND OF THE INVENTION

Field of the Invention

[0003] The invention relates to delivering web pages and other objects from a server, using techniques such as predictive predownload and delta encoding.

Description of Related Art

[0004] When serving web pages and other objects from a web server, it is advantageous to substantially minimize the amount of time needed to send those objects from the server to the requesting client (also known as a browser).

[0005] One method includes attempting to maintain template information already known to the client, and to send only delta information from the server to the client. However, the template information must be determined and sent to the client after the client has requested a corresponding web page. This method suffers from the drawback that it does not substantially minimize the amount of time required to present the page to a client as because the template information, the delta information (or both) are not sent to the client as soon as possible.

[0006] Another method is to predownload web pages or other objects from the server to the client. This method includes selecting one or more next objects the client is likely to request, and sending those next objects from the server to the client in a predownload time before the client actually requests those next objects. However, this suffers from the drawback that if the selected next objects take a relatively long

time to predownload, there is a substantial chance they might not be fully received before the client makes its next request.

BRIEF SUMMARY OF THE INVENTION

[0007] The invention includes a method and system which substantially minimizes the time needed to send and present objects from a server to a client, such as by using techniques for predictive predownload of templates with delta encoding. In one embodiment, a template builder generates a set of templates for each web page or other object. A prediction engine maintains a prediction map, responsive to web pages and other objects, the objects including the templates for web pages. The prediction engine selects one or more next objects likely to be requested by the client making the particular request, such as a next object or an object embedded in or referenced by a page. A delta encoder for a web page determines delta information in response to a current version of that page, and template information for that page, and encodes the web page for delivery to the client using the template information and delta information. The client is able to present the object in response to both the template information and delta information.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] Figure 1 shows a block diagram of a system including techniques involving predictive predownload of templates with delta caching.

[0009] Figure 2 shows a process flow diagram of a method of operation of a system including techniques involving predictive predownload of templates with delta caching.

DETAILED DESCRIPTION OF THE INVENTION

[0010] The description herein includes a preferred embodiment of the invention, including preferred data structures and process steps. Those skilled in the art would realize after perusal of this application, that embodiments of the invention might be implemented using a variety of other techniques not necessarily specifically described herein, without undue experimentation or further invention, and that such other techniques would be within the concept, scope, and spirit of the invention.

Lexicography

[0011] The following terms relate or refer to aspects of the invention or its embodiments. The general meaning of each of these terms is intended to be illustrative and in no way limiting.

[0012] **client** and **server** — These terms refer to a relationship between two elements in a system (whether actual hardware devices, software elements, or some combination thereof), particularly to their relationship as client and server, not necessarily to any particular physical devices.

[0013] For example, but without limitation, a particular client (whether a hardware device or a software element) in a first relationship with a first server, can also be a server in a second relationship with a second client.

[0014] **client device** and **server device** — These terms refer to devices taking on the role of a client or a server in a client-server relationship (such as an HTTP web client and web server). There is no particular requirement that any client devices or server devices must be individual physical devices, or even that they must be hardware entities. They can each be a single device, a set of cooperating devices, a portion of a device, a single software element, a set of cooperating software elements, a portion of a software element, or some combination thereof.

[0015] **web server** — This refers to a server capable of providing web objects, including web pages and data elements embedded therein, to requesting clients.

[0016] **delivery** — in general, sending a web page from a web server to a web client.

[0017] **delta information, delta encoding**, encoding **for delivery** — as used herein, the term "delta information" refers to a selected portion of a web page that may vary between instances of the web page. Delta encoding and encoding for delivery refer to a process wherein a device determines template information and delta information, provides mapping functions between template URLS and server URLs, delivers information that will be subsequently integrated into a web page or other document for delivery to an end user and provides a transparent interface between a client device and a server.

[0018] **object embedded in** or **referenced by** a web page — in general, an object embedded in or referenced by a web page includes links to other information such as other web pages.

[0019] **prediction engine, prediction map** — in general, a prediction engine selects one or more objects likely to be requested by the client making the particular request, such as a next page or an object embedded in or referenced by a page. A prediction map, responsive to web pages and other objects is maintained by the prediction engine.

[0020] **pre-download** — in general, to download an object to a client device at a point in time before the object is requested by a client.

[0021] **template information, template builder** – in general, the term "template information" refers to a selected portion of a web page that is relatively unchanging. If there is no difference between different instances of a web page, then the entire page may composed of template information. A "template builder" refers to a technique for determining template information. De-coupling the service of template information from the service of delta information to a client increases the overall speed of serving the web page.

[0022] **web objects** – in general, web pages, data elements embedded in web page and elements of web pages such as template information and delta information.

[0023] The scope and spirit of the invention is not limited to any of these definitions, or to specific examples mentioned therein, but is intended to include the most general concepts embodied by these and other terms.

System Elements

[0024] Figure 1 shows a block diagram of a system including techniques involving predictive predownload of templates with delta encoding.

[0025] A system 100 includes a set of clients 110, a communication network 120, a server 130, and a predownloader 140. Although the system 100 is described herein with regard to a single client 110 and a single server 130, those of ordinary skill in the art would understand, after perusal of this application, that the system 100 can also include multiple clients 110, multiple servers 130, or some combination thereof.

Moreover, although the system 100 is described with regard to requests for web objects, such as using the HTTP protocol (hypertext transfer protocol) or a variant thereof, those of ordinary skill in the art would understand, after perusal of this application, that the system 100 can also or alternatively include other client/server relationships, and requests for other types of objects.

[0026] There is no particular requirement that clients 110 are all of the same type or servers 130 are all of the same type. For one example, some clients 110 might use Microsoft's "Internet Explorer" web browser, while other clients 110 might use the open source "Opera" web browser. For another example, some servers 130 might use the open source "Apache" web server, while other servers 130 might use another web server.

[0027] There is no particular requirement that objects being requested, or protocols used to request them, are all of the same type. For example, web objects might be requested using FTP (file transfer protocol), HTTP, or variants thereof. File objects might be requested using FTP, NFS (network file system), or variants thereof. Database objects might be requested using SQL (structured query language), CORBA (common object request broker architecture), or variants thereof.

[0028] Each web client 110 includes a processor, program and data memory, mass storage, input elements (such as a keyboard or a mouse or other pointing device) and output elements (such as a monitor or other display and a speaker), and a client cache 111, and is controlled by a user 112. The processor, program and data memory, and mass storage operate in conjunction to perform the functions of a web client 110 (also known as a web "browser"). The web client 110 generates outgoing messages 113 including requests for web objects, which it sends to the web server 130, and receives incoming messages 113 including responses to those requests, which can include data from web objects. In one embodiment, the web client 110 uses the HTTP protocol or a variant thereof, but as noted above, other protocols can also or alternatively be used. The client cache 111 includes a portion of the data memory or mass storage, and is capable of storing copies of data from web objects.

[0029] Each web client 110 is coupled to the communication network 120. In one embodiment, the communication network 120 includes at least a portion of an

Internet, intranet, extranet, virtual private network, enterprise network, another form of communication network, or any other network, system, or technique capable of routing messages between and among one or more web clients 110 and web servers 130.

[0030] Each server 130 includes a processor, program and data memory, and mass storage. The data memory and mass storage are capable of storing objects 131, such as web pages, text, images, sounds, programs or program fragments, style-sheets, scripts, and other forms of data. Each object 131 can include one or more links 132 to other objects 131 (such as other web pages), and can also or alternatively include one or more embedded objects (such as embedded images, scripts, or other data). One or more of the servers 130 can also include input elements (such as a keyboard or a mouse or other pointing device) and output elements (such as a monitor or other display and a speaker), and be controlled by an operator 133.

[0031] The processor, program and data memory, and mass storage operate in conjunction to perform the functions of a server 130. In one embodiment, one or more of the servers 130 includes a plurality of devices each capable of acting as a server, cooperating using a load-sharing or other distribution technique for dividing the work of responding to requests among them, and jointly controlled by a single operator 133. Where multiple devices operate together, the server 130 can also be referred to as a "server farm".

[0032] The server 130 receives incoming messages 113 including requests for objects, and generates outgoing messages 113 including responses to those requests, which can include data from web objects. In one embodiment, the server 130 uses the HTTP protocol or a variant thereof, but as noted above other protocols can also or alternatively be used.

[0033] The predownloader 140 includes a delta encoder 141, a template builder 142, a predictor 143, a set of template information 144, and a set of prediction information 145. Although the predownloader 140 is described herein as a single device, those of ordinary skill in the art would recognize, after perusal of this application, that the predownloader 140 can be embodied as one or more devices, and including one or

more elements such as the delta encoder 141, the template builder 142, and the predictor 143.

[0034] There is no particular requirement that the delta encoder 141, the template builder 142, and the predictor 143 are embodied in the same device or element. There can be a separate device or element for each of them or some of them. For example, in one embodiment, the delta encoder 141 and the predictor 143 can be embodied in a first device, while the template builder 142 is embodied in a second device.

[0035] There is no particular requirement that the delta encoder 141, the template builder 142, and the predictor 143 are each single devices or elements. There can be a multiple devices or elements, or some combination thereof, for each of them or some of them. For example, in one embodiment, there can be multiple delta encoders 141, multiple predictors 143, and one template builder 142.

[0036] There is no particular requirement that the template information 144 or the prediction information 145 are recorded in single databases. There can be a multiple copies of either or both of them, or there can be separate (and possibly different) copies of either or both of them, or some combination thereof. For example, in one embodiment, there can be multiple copies of the same template information 144 collectively used by different servers 130 (with similar data), while there is a separate set of prediction information 145 for each server 130.

[0037] From the server 130, the delta encoder 141 receives data for an object 131 to be delivered to a client 110. The delta encoder 141 determines if there is a stored template for the object 131 in the template information 144. The delta encoder 141 calculates delta information for the object 131, responsive to the object 131 and responsive to the template information 144. The delta encoder 141 generates a message 113 to the client 110 including the delta information and specifying the template it used. The client 110 is capable of presenting the object 131 in response to the message 113, by reconstructing the object 131 in response to the delta information and a template in the client cache 111. If the client cache 111 does not contain the template, the client 110 is capable of receiving that template from the server 130 and then reconstructing the object 131.

[0038] The delta encoder 141 calculates the size of the delta information it calculates for each object 131, and determines in response to that size, in response to a size for the object 131 itself, and in response to a threshold value (possibly set by the operator 133) whether the delta information is larger than it "should be." If so, the delta encoder 141 so informs the template builder 142 (using any available technique), thus allowing the template builder 142 to calculate a new template for the object 131. There is no particular requirement for the delta encoder 141 to use any particular technique for so informing the template builder 142; it can use a flag or other information in a database record (including for example marking the template "obsolete"), an interprocess message (including a remote procedure call in systems 100 where the delta encoder 141 and the template builder 142 are not executing on the same hardware device), a network message (such as a message 113 in the system 100), or any other technique capable of allowing the template builder 142 to understand the results of the calculation by the delta encoder 141.

[0039] The template builder 142 calculates templates for selected objects 131 and records one or more of those templates for each of the selected objects 131 in the template information 144. Thus, the template information 144 can include multiple templates for one selected object 131. For example, if object 131 has been changed more than once recently. Alternatively, template information 144 can include one template for a group of more than one selected object 131 when those objects 131 have similar information. In one embodiment, the template builder 142 marks each object with a calculated template with a pointer to that template or other indicator for that template.

[0040] In a first embodiment, the template builder 142 calculates templates in response to information from the delta encoder 141, indicating that the delta information for a selected object 131 is larger than it "should be." In a second embodiment, the template builder 142 calculates templates in response to changes in selected objects 131 as each such object 131 is changed at the server 130. In a third embodiment, the template builder 142 calculates templates for all selected objects 131 in a sweep across the entire set of such objects 131 maintained by the server

130. Selected objects 131 can include all objects 131 at the server 130, or a subset thereof, where the subset is determined in response to one or more of the following:

[0041] **object age** — Templates are calculated for objects 131 changed more recently than, or which are older than, a selected threshold (possibly set by the operator 133), or some combination thereof.

[0042] **object size** — Templates are calculated for objects 131 larger than, or smaller than, a selected threshold (possibly set by the operator 133), or some combination thereof.

[0043] **object type** — Templates are calculated for objects 131 having one of a first set of selected types, or not having one of a second set of selected types (possibly set by the operator 133), or some combination thereof. For just one example, templates might be calculated for all text objects, but for no sound objects.

[0044] **operator selection** — Templates are calculated for objects 131 selected by the operator 133, individually or in groups.

[0045] Those of ordinary skill in the art would recognize, after perusal of this application, that any one or more of these embodiments, or some combination thereof, might be used in the system 100.

[0046] The predictor 143 receives the identity of objects 131 requested by the client 110 (such as by receiving copies of the messages 113 requesting those objects 131). In one embodiment, each such message 113 includes a "referring page," indicating the object 131 just previously requested by the client 110. Using aggregates of this information, the predictor 143 constructs a directed graph of likelihood information, indicating for each object 131, what next objects 131 are most likely and what is their relative likelihood. The predictor 143 can also or alternatively operate by predicting next objects 131 in response to any links 132 present in each object 131, in response to aggregate information for which objects 131 are most likely to be requested independent of the referring page.

[0047] As noted above, if a client 110 receives delta information but does not have corresponding template information, that client 110 can request the associated template from the server 130. The associated template includes a script that can be

executed by the client 110, and therefore can be treated by the predictor 143 like any other next object 131. In systems 100 where the associated template does not include a script (such as where delta encoding is performed using an extension or modification of the HTTP protocol), templates for each object 131 are still treated as next objects 131 that can be next requested by clients 110, and whose likelihood of being requested is calculated by the predictor 143.

[0048] In response to the predictor 143, the client 110 and the server 130 cooperate to pre-download (for example, to download to the client 110 and store in the client cache 111 before the user 112 specifically requests that object 131) the template for the object 131. In a first embodiment, the predictor 143 embeds a "hint" in the response message 113 to the client 110, suggesting a next object 131 to request if there is any idle time before the user specifically requests an object 131. In a second embodiment, the server 130 can generate a message 113 delivering the predicted next object 131, independently of an explicit request, which the client 110 stores in the client cache 111. Those of ordinary skill in the art, after perusal of this application, would recognize that the client 110 and the server 130 can operate in many different ways to pre-download the predicted next object 131 to the client 110 and store it in the client cache 111, and that all such techniques are within the scope and spirit of the invention.

[0049] In some embodiments, the client 110 and the server 130 can cooperate to deliver a compressed version of the delta information, the template information, or both, from the server 130 to the client 110. Delivering a compressed version of an object 131 can include either an extension or modification of the HTTP protocol, or can include other techniques. The system 100 can flexibly perform compression, including compressing only the delta information, only the template information, or compressing each separately.

Method of Operation

[0050] Figure 2 shows a process flow diagram of a method of operation of a system including techniques involving predictive predownload of templates with delta caching.

[0051] A method 200 is performed by the system 100. Although the method 200 is described serially, the steps of the method 200 can be performed by separate elements in conjunction or in parallel, whether asynchronously, in a pipelined manner, or otherwise. There is no particular requirement that the method 200 be performed in the same order in which this description lists the steps, except where so indicated.

[0052] At a flow point 210, predownloader 140 is ready to receive a request for an object 131.

[0053] At a step 211, the predownloader 140 receives a request message 113 from a client 110 for an object 131 from the server 130. The request message 113 includes a requested URL.

[0054] At a step 212, the predownloader 140 provides the requested URL to the delta encoder 141.

[0055] At a step 213, the predownloader 140 receives a referring-page URL, included in the request message 113.

[0056] At a step 214, the predownloader 140 provides the referring-page URL and the requested URL to the predictor 143.

[0057] At a step 221, the delta encoder 141 receives the requested URL included in the request message 113 from the predownloader 140.

[0058] At a step 222, the delta encoder 141 searches template information 144 and determines if there is a template for the object 131.

[0059] At a step 223, the delta encoder 141 calculates delta information for the object 131.

[0060] At a step 224, the delta encoder 141 determines whether the delta information for the object 131 is too large. In the event that the delta information for object 131 is too large, the delta encoder 141 informs the template builder 142 that the delta information for the object 131 is too large and a new template is generated.

[0061] At a step 225, the delta encoder 141 attaches any predownload hint information for the object 131 from the prediction information 145.

[0062] At a step 231, the template builder 142 calculates a template for the object 131 (if needed), and records that template in the template information 144.

[0063] At a step 241, the predictor 143 receives the referring-page URL and the requested URL from the predownloader 140. The referring-page URL and the requested URL are included in the message 113.

[0064] At a step 242, the predictor 143 calculates, in response to the referring-page URL and the requested URL, the likeliest next objects 131 for the object 131. The likeliest next objects include those objects associated with the object 131 that the user 112 is likely to request.

[0065] At a step 243, the predictor 143 updates the prediction information for the object 131, including predownload hint information for the object 131.

[0066] At a flow point 250, the predownloader 140 is done with the request. The method 200 continues at the flow point 210, where it is ready to receive another request.

Alternative Embodiments

[0067] Although preferred embodiments are disclosed herein, many variations are possible that remain within the concept, scope, and spirit of the invention. These variations would become clear to those skilled in the art after perusal of this application.

[0068] Those skilled in the art would realize that any alternative embodiments described herein are illustrative, and in no way limiting.